```
                                    B 7

NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
```

D 4

```
**FILE**ID**CNFSTORE

  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  TTTTTTTTTT    OOOOOO    RRRRRRRR    EEEEEEEEEE
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  TTTTTTTTTT    OOOOOO    RRRRRRRR    EEEEEEEEEE
CC          NN      NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NN      NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NNNN    NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NNNN    NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NN  NN  NN  FFFFFFFF      SSSSSS        TT      OO      OO  RRRRRRRR    EEEEEEEE
CC          NN  NN  NN  FFFFFFFF      SSSSSS        TT      OO      OO  RRRRRRRR    EEEEEEEE
CC          NN    NNNN  FF                  SS      TT      OO      OO  RR  RR      EE
CC          NN    NNNN  FF                  SS      TT      OO      OO  RR  RR      EE
CC          NN      NN  FF                  SS      TT      OO      OO  RR    RR    EE              ....
CC          NN      NN  FF                  SS      TT      OO      OO  RR    RR    EE              ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS        TT        OOOOOO    RR      RR  EEEEEEEEEE      ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS        TT        OOOOOO    RR      RR  EEEEEEEEEE      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0001   0 %TITLE  'DECnet Ethernet Configurator Module'
0002   0 MODULE CNFSTORE                 (
0003   0                                 LANGUAGE (BLISS32),
0004   0                                 IDENT = 'V04-000'
0005   0                                 ) =
0006   1 BEGIN
0007   1
0008   1 !
0009   1 !*************************************************************************
0010   1 !*                                                                       *
0011   1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
0012   1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
0013   1 !*  ALL RIGHTS RESERVED.                                                 *
0014   1 !*                                                                       *
0015   1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0016   1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0017   1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0018   1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0019   1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0020   1 !*  TRANSFERRED.                                                         *
0021   1 !*                                                                       *
0022   1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0023   1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0024   1 !*  CORPORATION.                                                         *
0025   1 !*                                                                       *
0026   1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0027   1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
0028   1 !*                                                                       *
0029   1 !*                                                                       *
0030   1 !*************************************************************************
0031   1 !
0032   1
0033   1 !++
0034   1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
0035   1 !
0036   1 ! ABSTRACT:
0037   1 !
0038   1 !       This module contains the routines for reading and storing the
0039   1 !       system ID messages that are periodically broadcast on the NI.
0040   1 !
0041   1 ! ENVIRONMENT:  VAX/VMS Operating System
0042   1 !
0043   1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  13-Oct-1982
0044   1 !
0045   1 ! MODIFIED BY:
0046   1 !
0047   1 !       V03-002 TRC0002         Terry Cassidy           Aug-23-1984
0048   1 !               Inhibit processing of pad bytes in Sys ID
0049   1 !
0050   1 !       V03-001 RPG0001         Bob Grosso              May-19-1983
0051   1 !               Correct the arguements to LIB$SIGNAL.
0052   1 !
0053   1 !--
```

```
;    55          0054   1 %SBTTL  'Definitions'
;    56          0055   1
;    57          0056   1 !
;    58          0057   1 ! INCLUDE FILES:
;    59          0058   1 !
;    60          0059   1
;    61          0060   1 LIBRARY 'SYS$LIBRARY:STARLET';                         ! VMS common definitions
;    62          0061   1
;    63          0062   1 LIBRARY 'SHRLIB$:NMALIBRY';                            ! NICE code definitions
;    64          0063   1
;    65          0064   1 REQUIRE 'LIB$:CNFDEF.R32';
;    66          0155   1
;    67          0156   1 REQUIRE 'SRC$:CNFPREFIX.REQ';
;    68          0253   1
;    69          0254   1
;    70          0255   1 !
;    71          0256   1 ! BUILTIN functions
;    72          0257   1 !
;    73          0258   1
;    74          0259   1 BUILTIN
;    75          0260   1      INSQUE,                         ! INSQUE instruction
;    76          0261   1      REMQUE;                         ! REMQUE instruction
;    77          0262   1
;    78          0263   1 GLOBAL LITERAL
;    79          0264   1      SYSIDM_BUFSIZ = 64,
;    80          0265   1      ADRTYP_BUFSIZ = 14;
;    81          0266   1
;    82          0267   1 !
;    83          0268   1 ! TABLE OF CONTENTS:
;    84          0269   1 !
;    85          0270   1
;    86          0271   1 FORWARD ROUTINE
;    87          0272   1
;    88          0273   1           CNF$READ_SYSIDM,           ! Read the system ID messages from the NI
;    89          0274   1           BUFFER_ID : NOVALUE,       ! Buffer the ID messages and re-issue read
;    90          0275   1           STORE_ID : NOVALUE,        ! Partially parse the messages and store them
;    91          0276   1           INSERT_SID;                ! Place the ID messages in the circuit block
;    92          0277   1
;    93          0278   1 !
;    94          0279   1 ! EXTERNAL REFERENCES:
;    95          0280   1 !
;    96          0281   1
;    97          0282   1 EXTERNAL ROUTINE
;    98          0283   1
;    99          0284   1 !    Module CNFMAIN
;   100          0285   1
;   101          0286   1           CNF$EXIT,                  ! Clean up and exit
;   102          0287   1           CNF$TRACE,                 ! Log messages to log file
;   103          0288   1           CNF$GET_ZVM,               ! Get zeroed virtual memory
;   104          0289   1           CNF$FREE_VM,               ! Free virtual memory
;   105          0290   1
;   106          0291   1 !    Module CNFWORKQ
;   107          0292   1
;   108          0293   1           WKQ$ADD_WORK_ITEM;         ! Add work to work queue
;   109          0294   1
;   110          0295   1 EXTERNAL ROUTINE
;   111          0296   1
```

```
1               STR$COMPARE     : ADDRESSING_MODE (GENERAL);
1
1
1       EXTERNAL LITERAL
1
1               CNF$C_ASYNCH_EFN;
1
1
1       EXTERNAL LITERAL
1
1               CNF$_LOGIC,     ! Program logic error or unexpected condition
1               CNF$_SYSID;     ! Error while obtaining system ID message
1
1       EXTERNAL
1
1               CNF$GQ_CIRSURLST : VECTOR [2];  ! List of circuits under surveillance
1
```

```
;  130          0314  1 %SBTTL  'CNF$READ_SYSIDM  Read the system id messages on the NI'
;  131          0315  1 GLOBAL ROUTINE CNF$READ_SYSIDM (CIR) =
;  132          0316  1
;  133          0317  1 !++
;  134          0318  1 ! FUNCTIONAL DESCRIPTION:
;  135          0319  1 !
;  136          0320  1 !   CNF$READ_SYSIDM issues the QIO to read System ID Messages on the NI.
;  137          0321  1 !
;  138          0322  1 ! FORMAL PARAMETERS:
;  139          0323  1 !
;  140          0324  1 !       cir     CIRcuit control block.  Contains buffers and information
;  141          0325  1 !               on the circuit.
;  142          0326  1 !
;  143          0327  1 ! IMPLICIT INPUTS:
;  144          0328  1 !       NONE
;  145          0329  1 !
;  146          0330  1 ! IMPLICIT OUTPUTS:
;  147          0331  1 !       NONE
;  148          0332  1 !
;  149          0333  1 ! ROUTINE VALUE:
;  150          0334  1 ! COMPLETION CODES:
;  151          0335  1 !       Always return success
;  152          0336  1 !
;  153          0337  1 ! SIDE EFFECTS:
;  154          0338  1 !       NONE
;  155          0339  1 !
;  156          0340  1 !--
;  157          0341  1
;  158          0342  2     BEGIN
;  159          0343  2     MAP
;  160          0344  2         CIR : REF BBLOCK;
;  161          0345  2     LOCAL
;  162          0346  2         STATUS;
;  163          0347  2
;  164          0348  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  165          0349  2         $DESCRIPTOR ('cnf$read_sysidm'));
;  166          0350  2
;  167          0351  2     !
;  168          0352  2     !   Allocate the two buffers to hold the system ID message
;  169          0353  2     !   and the message header.  If the buffers are already there
;  170          0354  2     !   then a logic error of some sort has occurred.
;  171          0355  2     !
;  172          0356  2     IF  (.CIR [CIR$L_SYSIDMBUF] NEQ 0) OR
;  173          0357  3         (.CIR [CIR$L_ADRTYPBUF] NEQ 0)
;  174          0358  2     THEN
;  175          0359  2         SIGNAL (CNF$_LOGIC);
;  176          0360  2     EXECUTE (CNF$GET_ZVM (%REF(SYSIDM_BUFSIZ), CIR [CIR$L_SYSIDMBUF]) );
;  177          0361  2     EXECUTE (CNF$GET_ZVM (%REF(ADRTYP_BUFSIZ), CIR [CIR$L_ADRTYPBUF]) );
;  178          0362  2
;  179          0363  2
;  180          0364  2     CH$FILL (0, 8, CIR [CIR$W_IOSB]);   ! Initialize the I/O status block
;  181          0365  2
;  182          0366  2 !
;  183          0367  2 !   Issue QIO to read system id messages being broadcast on the NI
;  184          0368  2 !
;  185        P 0369  2     STATUS = $QIO
;  186        P 0370  2         (
;  187       P 0371  2             FUNC = IO$_READLBLK,
;  188       P 0372  2             CHAN = .CIR [CIR$W_CHAN],
;  189       P 0373  2             EFN = CNF$C_ASYNCH_EFN,
;  190       P 0374  2             IOSB = CIR [CIR$W_IOSB],
;  191       P 0375  2             ASTADR = BUFFER_ID,
;  192       P 0376  2             ASTPRM = .CIR,
;  193       P 0377  2             P1 = .CIR [CIR$L_SYSIDMBUF],
;  194       P 0378  2             P2 = SYSIDM_BUFSIZ,
;  195       P 0379  2             P5 = .CIR [CIR$L_ADRTYPBUF]
;  196         0380  2             );
;  197         0381  2
;  198         0382  2        IF NOT .STATUS THEN SIGNAL (CNF$_SYSID, 0, .STATUS);
;  199         0383  2        RETURN TRUE;
;  200         0384  1        END;                                     ! routine CNF$READ_SYSIDM


                                                               .TITLE  CNFSTORE DECnet Ethernet Configurator Module
                                                               .IDENT  \V04-000\

                                                               .PSECT  $PLIT$,NOWRT,NOEXE,2

                          45  43  41  52  54  00000 P.AAB:    .ASCII  \TRACE\                                       ;
                                              00005           .BLKB   3
                                   00000005  00008 P.AAA:    .LONG   5                                             ;
                                   00000000' 0000C           .ADDRESS P.AAB                                        ;
6D  64  69  73  79  73  5F  64  61  65  72  24  66  6E  63  00010 P.AAD:    .ASCII  \cnf$read_sysidm\                  ;
                                              0001F           .BLKB   1
                                   0000000F  00020 P.AAC:    .LONG   15                                            ;
                                   00000000' 00024           .ADDRESS P.AAD                                        ;

                                                     SYSIDM_BUFSIZ==     64
                                                     ADRTYP_BUFSIZ==     14
                                                               .EXTRN  CNF$EXIT, CNF$TRACE
                                                               .EXTRN  CNF$GET_ZVM, CNF$FREE_VM
                                                               .EXTRN  WKQ$ADD_WORK_ITEM
                                                               .EXTRN  STR$COMPARE, CNF$C_ASYNCH_EFN
                                                               .EXTRN  CNF$_LOGIC, CNF$_SYSID
                                                               .EXTRN  CNF$GQ_CIRSURLST
                                                               .EXTRN  SYS$QIO

                                                               .PSECT  $CODE$,NOWRT,2

                                         00FC 00000           .ENTRY  CNF$READ_SYSIDM, Save R2,R3,R4,R5,R6,R7      ; 0315
                     57 00000000G    00   9E 00002           MOVAB   LIB$SIGNAL, R7                                ;
                     5E              04   C2 00009           SUBL2   #4, SP                                        ;
                             0000'   CF   9F 0000C           PUSHAB  P.AAC                                         ; 0349
                             0000'   CF   9F 00010           PUSHAB  P.AAA                                         ; 0348
                                     01   DD 00014           PUSHL   #1                                            ;
                0000G   CF           03   FB 00016           CALLS   #3, CNF$TRACE                                 ;
                        56       04  AC   D0 0001B           MOVL    CIR, R6                                       ; 0356
                                 38  A6   D5 0001F           TSTL    56(R6)                                        ;
                                     05   12 00022           BNEQ    1$                                            ;
                                 3C  A6   D5 00024           TSTL    60(R6)                                        ; 0357
                                     09   13 00027           BEQL    2$                                            ;
                          00000000G  8F   DD 00029 1$:       PUSHL   #CNF$_LOGIC                                   ; 0359
                        67           01   FB 0002F           CALLS   #1, LIB$SIGNAL                                ;
                                 38  A6   9F 00032 2$:       PUSHAB  56(R6)                                        ; 0360
```

```
                       04  AE         40  8F  9A 00035         MOVZBL  #64, 4(SP)                ;
                                      04  AE  9F 0003A         PUSHAB  4(SP)                     ;
                    0000G  CF             02  FB 0003D         CALLS   #2, CNF$GET_ZVM           ;
                           56             50  E9 00042         BLBC    STATUS, 4$                ;
                                      3C  A6  9F 00045         PUSHAB  60(R6)                    ; 0361
                       04  AE             0E  D0 00048         MOVL    #14, 4(SP)                ;
                                      04  AE  9F 0004C         PUSHAB  4(SP)                     ;
                    0000G  CF             02  FB 0004F         CALLS   #2, CNF$GET_ZVM           ;
                           44             50  E9 00054         BLBC    STATUS, 4$                ;
        08          00     6E             00  2C 00057         MOVC5   #0, (SP), #0, #8, 12(R6)  ; 0364
                                      0C  A6     0005C                                           ;
                                          7E  D4 0005E         CLRL    -(SP)                     ; 0380
                                      3C  A6  DD 00060         PUSHL   60(R6)                    ;
                                          7E  7C 00063         CLRQ    -(SP)                     ;
                           7E         40  8F  9A 00065         MOVZBL  #64, -(SP)                ;
                                      38  A6  DD 00069         PUSHL   56(R6)                    ;
                                          56  DD 0006C         PUSHL   R6                        ;
                                   0000V  CF  9F 0006E         PUSHAB  BUFFER_ID                 ;
                                      0C  A6  9F 00072         PUSHAB  12(R6)                    ;
                                          21  DD 00075         PUSHL   #33                       ;
                           7E         14  A6  32 00077         CVTWL   20(R6), -(SP)             ;
                               00000000G  8F  DD 0007B         PUSHL   #CNF$C_ASYNCH_EFN         ;
                00000000G  00             0C  FB 00081         CALLS   #12, SYS$QIO              ;
                           0D             50  E8 00088         BLBS    STATUS, 3$                ; 0382
                                          50  DD 0008B         PUSHL   STATUS                    ;
                                          7E  D4 0008D         CLRL    -(SP)                     ;
                               00000000G  8F  DD 0008F         PUSHL   #CNF$_SYSID               ;
                           67             03  FB 00095         CALLS   #3, LIB$SIGNAL            ;
                           50             01  D0 00098 3$:     MOVL    #1, R0                    ; 0383
                                              04 0009B 4$:     RET                               ; 0384

; Routine Size:  156 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'buffer_id   Remove the system id message from buffer'
1 ROUTINE BUFFER_ID (CIR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !   Buffer_id is an AST routine called when a read for a system ID on
1 !   the NI completes or cancels.  If the read was cancelled then the
1 !   circuit's buffers are deallocated.  If the read was successful
1 !   then the id message must be removed to another buffer so that the buffer
1 !   can be re-used and the read re-issued immediately.  The circuit block
1 !   and the id message will be queued to the work queue to be parsed and stored.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       cir     CIRcuit control block.  Contains buffers and information
1 !               on the circuit.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2     BEGIN
2     MAP
2         CIR : REF BBLOCK;
2
2     LOCAL
2         SYSIDM : REF BBLOCK;
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('buffer_id'));
2
2     !
2     !   Check to see if surveillance has been disabled.
2     !   Since surveillance is disabled by a routine executed from
2     !   the work queue, and this routine is delivered by AST,
2     !   care must be taken to ensure that errors are not introduced
2     !   due to the timing of the AST completion.
2     !
2
2     !
2     !   If the circuit was marked for disable but the AST was delivered
2     !   before the channel was deassigned, then just quit.
2     !
2     IF .CIR [CIR$B_SURVEIL] EQL NMA$C_SUR_DIS
2     THEN
```

```
;   259         0442  3         BEGIN
;   260         0443  3         EXECUTE (CNF$FREE_VM (%REF(ADRTYP_BUFSIZ), CIR [CIR$L_ADRTYPBUF]));
;   261         0444  3         EXECUTE (CNF$FREE_VM (%REF(SYSIDM_BUFSIZ), CIR [CIR$L_SYSIDMBUF]));
;   262         0445  3         CIR [CIR$L_ADRTYPBUF] = 0;
;   263         0446  3         CIR [CIR$L_SYSIDMBUF] = 0;
;   264         0447  3         RETURN TRUE;
;   265         0448  2         END;
;   266         0449  2
;   267         0450  2     !
;   268         0451  2     !   The I/O failed, and not because the surveillance was disabled
;   269         0452  2     !   and the I/O cancelled, therefore signal the error.
;   270         0453  2     !
;   271         0454  2     IF NOT .CIR [CIR$W_IOSB]
;   272         0455  2     THEN
;   273         0456  3         BEGIN
;   274         0457  3         SIGNAL (CNF$_SYSID, 0, .CIR [CIR$W_IOSB]);
;   275         0458  3         RETURN TRUE;
;   276         0459  2         END;
;   277         0460  2
;   278         0461  2     !
;   279         0462  2     !   Allocate a descriptor and fill it in
;   280         0463  2     !
;   281         0464  2     EXECUTE (CNF$GET_ZVM (%REF(DSC$C_S_BLN), SYSIDM));
;   282         0465  2
;   283         0466  2     SYSIDM [DSC$A_POINTER] = .CIR [CIR$L_SYSIDMBUF];
;   284         0467  2     SYSIDM [DSC$W_LENGTH] = .CIR [CIR$W_IOSB1];
;   285         0468  2
;   286         0469  2     !
;   287         0470  2     !   Do the work of copying the ID out of the CIR block at the
;   288         0471  2     !   the leisure of the work queue.  STORE_ID will have to check
;   289         0472  2     !   whether surveillance was disabled on the circuit between now
;   290         0473  2     !   and when it finally executes.
;   291         0474  2     !
;   292         0475  2     WKQ$ADD_WORK_ITEM (STORE_ID, .CIR, .SYSIDM, .CIR [CIR$L_ADRTYPBUF]);
;   293         0476  2
;   294         0477  2     CIR [CIR$L_SYSIDMBUF] = 0;
;   295         0478  2     CIR [CIR$L_ADRTYPBUF] = 0;
;   296         0479  2
;   297         0480  2     !
;   298         0481  2     !   While still at AST level, hurry up and re-issue the READ for the
;   299         0482  2     !   System ID messages to reduce the chance of missing any.
;   300         0483  2     !
;   301         0484  2     CNF$READ_SYSIDM (.CIR);
;   302         0485  2
;   303         0486  2     RETURN TRUE;
;   304         0487  1     END;                                ! Routine buffer_id


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                              45  43  41  52  54  00028 P.AAF:  .ASCII  \TRACE\                          ;
                                                  0002D         .BLKB   3
                                        00000005  00030 P.AAE:  .LONG   5                                ;
                                        00000000' 00034         .ADDRESS P.AAF                           ;
              64  69  5F  72  65  66  66  75  62  00038 P.AAH:  .ASCII  \buffer_id\                      ;
                                                  00041         .BLKB   3
```

```
                           00000009 00044 P.AAG:  .LONG    9                                ;
                          00000000' 00048        .ADDRESS P.AAH                            ;

                                                 .PSECT   $CODE$,NOWRT,2

                               0004 00000 BUFFER_ID:
                                                 .WORD    Save R2                          ; 0386
              5E             08  C2 00002        SUBL2    #8, SP                           ;
                      0000'  CF  9F 00005        PUSHAB   P.AAG                            ; 0426
                      0000'  CF  9F 00009        PUSHAB   P.AAE                            ; 0425
                             01  DD 0000D        PUSHL    #1                               ;
       0000G  CF             03  FB 0000F        CALLS    #3, CNF$TRACE                    ;
              52         04  AC  D0 00014        MOVL     CIR, R2                          ; 0440
              01         0A  A2  91 00018        CMPB     10(R2), #1                       ;
                             29  12 0001C        BNEQ     1$                               ;
                         3C  A2  9F 0001E        PUSHAB   60(R2)                           ; 0443
          04  AE             0E  D0 00021        MOVL     #14, 4(SP)                       ;
                         04  AE  9F 00025        PUSHAB   4(SP)                            ;
       0000G  CF             02  FB 00028        CALLS    #2, CNF$FREE_VM                  ;
              68             50  E9 0002D        BLBC     STATUS, 3$                       ;
                         38  A2  9F 00030        PUSHAB   56(R2)                           ; 0444
          04  AE         40  8F  9A 00033        MOVZBL   #64, 4(SP)                       ;
                         04  AE  9F 00038        PUSHAB   4(SP)                            ;
       0000G  CF             02  FB 0003B        CALLS    #2, CNF$FREE_VM                  ;
              55             50  E9 00040        BLBC     STATUS, 3$                       ;
                         38  A2  7C 00043        CLRQ     56(R2)                           ; 0446
                                 04 00046        RET                                       ; 0447
              14         0C  A2  E8 00047 1$:    BLBS     12(R2), 2$                       ; 0454
              7E         0C  A2  32 0004B        CVTWL    12(R2), -(SP)                    ; 0457
                             7E  D4 0004F        CLRL     -(SP)                            ;
                  00000000G  8F  DD 00051        PUSHL    #CNF$_SYSID                      ;
   00000000G  00             03  FB 00057        CALLS    #3, LIB$SIGNAL                   ;
                                 04 0005E        RET                                       ; 0458
                         04  AE  9F 0005F 2$:    PUSHAB   SYSIDM                           ; 0464
          04  AE             08  D0 00062        MOVL     #8, 4(SP)                        ;
                         04  AE  9F 00066        PUSHAB   4(SP)                            ;
       0000G  CF             02  FB 00069        CALLS    #2, CNF$GET_ZVM                  ;
              27             50  E9 0006E        BLBC     STATUS, 3$                       ;
              50         04  AE  D0 00071        MOVL     SYSIDM, R0                       ; 0466
          04  A0         38  A2  D0 00075        MOVL     56(R2), 4(R0)                    ;
              60         0E  A2  B0 0007A        MOVW     14(R2), (R0)                     ; 0467
                         3C  A2  DD 0007E        PUSHL    60(R2)                           ; 0475
                             50  DD 00081        PUSHL    R0                               ;
                             52  DD 00083        PUSHL    R2                               ;
                      0000V  CF  9F 00085        PUSHAB   STORE_ID                         ;
       0000G  CF             04  FB 00089        CALLS    #4, WRQ$ADD_WORK_ITEM            ;
                         38  A2  7C 0008E        CLRQ     56(R2)                           ; 0477
                             52  DD 00091        PUSHL    R2                               ; 0484
        FECC  CF             01  FB 00093        CALLS    #1, CNF$READ_SYSIDM              ;
                                 04 00098 3$:    RET                                       ; 0487

; Routine Size:  153 bytes,    Routine Base:  $CODE$ + 009C
;   306         0488  1 %SBTTL  'store_id   Parse and store the system id message'
;   307         0489  1 ROUTINE STORE_ID (CIR, SYSIDM_DSC, ADRTYPBUF) : NOVALUE =
;   308         0490  1
;   309         0491  1 !++
;   310         0492  1 ! FUNCTIONAL DESCRIPTION:
;   311         0493  1 !
;   312         0494  1 !     Store_id is executed off the work queue.
;   313         0495  1 !     The id message must be parsed and stored.
;   314         0496  1 !     If it is the id from a familiar device then the cell for
;   315         0497  1 !     that device's data is updated, otherwise a new cell is created.
;   316         0498  1 !
;   317         0499  1 ! FORMAL PARAMETERS:
;   318         0500  1 !
;   319         0501  1 !       cir             CIRcuit control block.  Contains buffers and information
;   320         0502  1 !                       on the circuit.
;   321         0503  1 !
;   322         0504  1 !       sysidm_dsc      Descriptor of buffer containing system ID message
;   323         0505  1 !
;   324         0506  1 !       adrtypbuf       Buffer containing the current NI address and
;   325         0507  1 !                       protocol type
;   326         0508  1 !
;   327         0509  1 ! IMPLICIT INPUTS:
;   328         0510  1 !       NONE
;   329         0511  1 !
;   330         0512  1 ! IMPLICIT OUTPUTS:
;   331         0513  1 !       NONE
;   332         0514  1 !
;   333         0515  1 ! ROUTINE VALUE:
;   334         0516  1 ! COMPLETION CODES:
;   335         0517  1 !       NONE
;   336         0518  1 !
;   337         0519  1 ! SIDE EFFECTS:
;   338         0520  1 !       NONE
;   339         0521  1 !
;   340         0522  1 !--
;   341         0523  1
;   342         0524  2     BEGIN
;   343         0525  2     MAP
;   344         0526  2         ADRTYPBUF : REF BBLOCK,
;   345         0527  2         CIR : REF BBLOCK,
;   346         0528  2         SYSIDM_DSC : REF BBLOCK;
;   347         0529  2     LOCAL
;   348         0530  2         SID : REF BBLOCK,               ! System Id storage block
;   349         0531  2         SIMBUF;
;   350         0532  2
;   351         0533  2
;   352         0534  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;   353         0535  2         $DESCRIPTOR ('store_id'));
;   354         0536  2
;   355         0537  2     !
;   356         0538  2     !   If the circuit was not marked for disable after the AST was delivered
;   357         0539  2     !   but before STORE_ID was called from the work queue then store the
;   358         0540  2     !   system ID message in the circuit block.
;   359         0541  2     !
;   360         0542  2     IF .CIR [CIR$B_SURVEIL] EQL NMA$C_SUR_ENA
;   361         0543  2     THEN
;   362         0544  3         BEGIN
```

```
3
3             !
3             !    Allocate a buffer to hold the system ID message.
3             !
3             EXECUTE (CNF$GET_ZVM (%REF(SID$C_LENGTH), SID) );
3             EXECUTE ($GETTIM ( TIMADR = SID [SID$Q_LSTREPORT] ));
3
3             !
3             !         Extract Current NI address and Protocol type from
3             !         the ADRTYPBUF buffer and deallocate the buffer.
3             !
3             CH$MOVE (SID$C_ADRLEN, .ADRTYPBUF + SID$C_ADRLEN, SID [SID$T_CURADR]);
3
3             !
3             !         If TYPE is not Protocol 260 (Remote Console system ID)
3             !         Then there's been an error.
3             !
3   !!        TBS
3
3             !
3             !         Extract and store info from the System ID Message buffer
3             !
3             SIMBUF = .SYSIDM_DSC [DSC$A_POINTER] + 4;        ! Point to beginning of buffer and skip past
3                                                              !  Code byte, Pad byte and Receipt number word
3             !
3             !    The system ID message is self describing.  A word of type code
3             !    is following by a word containing the length in bytes of the data
3             !    field which follows.  Loop reading a type word, length byte and
3             !    processing the data field, until the whole message has been
3             !    stored away in the appropriate fields of the SID which will be
3             !    placed in the CIR block.
3             !
3             WHILE .SIMBUF LSS (.SYSIDM_DSC [DSC$A_POINTER] + .SYSIDM_DSC [DSC$W_LENGTH]) DO
4                 BEGIN
4                 LOCAL
4                     LENGTH,
4                     TYPE;
4
4                 TYPE = .(.SIMBUF) <0, 16>;              ! Word of type code
4                 LENGTH = .(.SIMBUF) <16, 8>;            ! Byte of length of data field
4                 IF .LENGTH LEQ 0
4                 THEN
5                     BEGIN
5                     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
5                         $DESCRIPTOR ('store_id - illegal system ID field length'));
5                     EXITLOOP;
4                     END;
4
4                 SIMBUF = .SIMBUF + 3;                   ! Skip type and length to data
4
4                 !
4                 !    Dispatch on field type to store data in proper location in
4                 !    SID block.
4                 !
4                 SELECTONE .TYPE OF
4                     SET
```

```
:   420         0602  4                 !
:   421         0603  4                 !     MOP version, MOP version ECO, MOP version User ECO
:   422         0604  4                 !
:   423         0605  4                 [SIM$C_MOPVERTYP]:
:   424         0606  5                     BEGIN
:   425         0607  5                     IF .LENGTH EQL 3 THEN
:   426         0608  6                         BEGIN
:   427         0609  6                         SID [SID$B_MOPVER] = .(.SIMBUF) <0, 8>;           ! MOP version
:   428         0610  6                         SID [SID$B_MOPECO] = .(.SIMBUF) <8, 16>;          ! MOP version ECO
:   429         0611  6                         SID [SID$B_MOPUSRECO] = .(.SIMBUF) <16, 8>;       ! MOP version User ECO
:   430         0612  6                         END
:   431         0613  5                     ELSE
:   432         0614  5                         CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
:   433         0615  5                             $DESCRIPTOR ('store_id - MOP'));
:   434         0616  5                     SIMBUF = .SIMBUF + .LENGTH;                         ! Skip over data to next TYPE
:   435         0617  4                     END;
:   436         0618  4
:   437         0619  4                 !
:   438         0620  4                 !   Funtions are returned in a word bit mask.  To make
:   439         0621  4                 !   life easier when building the SHOW response later,
:   440         0622  4                 !   figure how many of the bits in the mask are set.
:   441         0623  4                 !
:   442         0624  4                 [SIM$C_FUNCTNTYP]:
:   443         0625  5                     BEGIN
:   444         0626  5                     BIND
:   445         0627  5                         FUNCTIONS = SID [SID$W_FUNCTIONS] : BITVECTOR [SID$C_MAXFUNC];
:   446         0628  5
:   447         0629  5                     IF .LENGTH NEQ 2
:   448         0630  5                     THEN
:   449         0631  5                         CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
:   450         0632  5                             $DESCRIPTOR ('store_id - FUNCTION LENGTH'));
:   451         0633  5                     SID [SID$W_FUNCTIONS] = .(.SIMBUF) <0, 16>;
:   452         0634  5                     SIMBUF = .SIMBUF + .LENGTH;                         ! Skip over data to next TYPE
:   453         0635  5
:   454         0636  5                     !
:   455         0637  5                     !   Count the number of bits that are set
:   456         0638  5                     !
:   457         0639  5                     INCR INDEX FROM 0 TO SID$C_MAXFUNC - 1 DO
:   458         0640  5                         IF .FUNCTIONS [.INDEX]
:   459         0641  5                         THEN SID [SID$B_NUMFUNC] = .SID [SID$B_NUMFUNC] + 1;
:   460         0642  4                     END;
:   461         0643  4
:   462         0644  4                 !
:   463         0645  4                 !   Get the 6-byte Hardware address
:   464         0646  4                 !
:   465         0647  4                 [SIM$C_HDWADRTYP]:
:   466         0648  5                     BEGIN
:   467         0649  5                     CH$MOVE (SID$C_ADRLEN, .SIMBUF, SID [SID$T_HRDWADR]);
:   468         0650  5                     SIMBUF = .SIMBUF + .LENGTH;                         ! Skip over data to next TYPE
:   469         0651  4                     END;
:   470         0652  4
:   471         0653  4                 !
:   472         0654  4                 !   Get the device type
:   473         0655  4                 !
:   474         0656  4                 [SIM$C_DEVICETYP]:
:   475         0657  5                     BEGIN
:   476         0658  5                     SID [SID$B_DEVICE] = .(.SIMBUF) <0, 8>;
```

```
;  477          0659  5                         SIMBUF = .SIMBUF + .LENGTH;                     ! Skip over data to next TYPE
;  478          0660  4                         END;
;  479          0661  4
;  480          0662  4                     !
;  481          0663  4                     !   There is a data field returned which we don't understand.
;  482          0664  4                     !   Skip over it and proceed merrily on our way.
;  483          0665  4                     !
;  484          0666  4                     [OTHERWISE]:
;  485          0667  5                         BEGIN
;  486          0668  5                         CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
;  487          0669  5                             $DESCRIPTOR ('store_id'));
;  488          0670  5                         SIMBUF = .SIMBUF + .LENGTH;                     ! Skip over data to next TYPE
;  489          0671  4                         END;
;  490          0672  4
;  491          0673  4                     TES;
;  492          0674  4
;  493          0675  3                 END;                                        ! While parsing System ID Message buffer
;  494          0676  3
;  495          0677  3             !
;  496          0678  3             !   Place the SID into the circuit block on an ordered linked list
;  497          0679  3             !
;  498          0680  3             INSERT_SID (.SID, .CIR);
;  499          0681  2             END;                                            ! If surveillance was still enabled
;  500          0682  2
;  501          0683  2         !
;  502          0684  2         !   If the circuit was marked for disable after the AST was delivered
;  503          0685  2         !   but before STORE_ID was called from the work queue then just
;  504          0686  2         !   deallocate the buffers and quit.
;  505          0687  2         !
;  506          0688  2         EXECUTE (CNF$FREE_VM (%REF(ADRTYP_BUFSIZ), ADRTYPBUF) );
;  507          0689  2         EXECUTE (CNF$FREE_VM (%REF(SYSIDM_BUFSIZ), SYSIDM_DSC [DSC$A_POINTER]) );
;  508          0690  2         EXECUTE (CNF$FREE_VM (%REF(DSC$C_S_BLN), SYSIDM_DSC));
;  509          0691  2
;  510          0692  2         RETURN TRUE;
;  511          0693  1         END;                                ! Routine Store_id


                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          45  43  41  52  54  0004C P.AAJ:  .ASCII  \TRACE\                                         ;
                                                              00051         .BLKB   3
                                                   00000005  00054 P.AAI:  .LONG   5                                               ;
                                                   00000000' 00058         .ADDRESS P.AAJ                                          ;
                              64  69  5F  65  72  6F  74  73  0005C P.AAL:  .ASCII  \store_id\                                      ;
                                                   00000008  00064 P.AAK:  .LONG   8                                               ;
                                                   00000000' 00068         .ADDRESS P.AAL                                          ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  0006C P.AAN:  .ASCII  \TRACE *** ERROR \                              ;
                                                          20  0007B                                                                 ;
                                                   00000010  0007C P.AAM:  .LONG   16                                              ;
                                                   00000000' 00080         .ADDRESS P.AAN                                          ;
65  6C  6C  69  20  2D  20  64  69  5F  65  72  6F  74  73  00084 P.AAP:  .ASCII  \store_id - illegal system ID field lengt\       ;
66  20  44  49  20  6D  65  74  73  79  73  20  6C  61  67  00093                                                                 ;
                    74  67  6E  65  6C  20  64  6C  65  69  000A2                                                                 ;
                                                          68  000AC         .ASCII  \h\                                             ;
                                                              000AD         .BLKB   3
                                                   00000029  000B0 P.AAO:  .LONG   41                                              ;
```

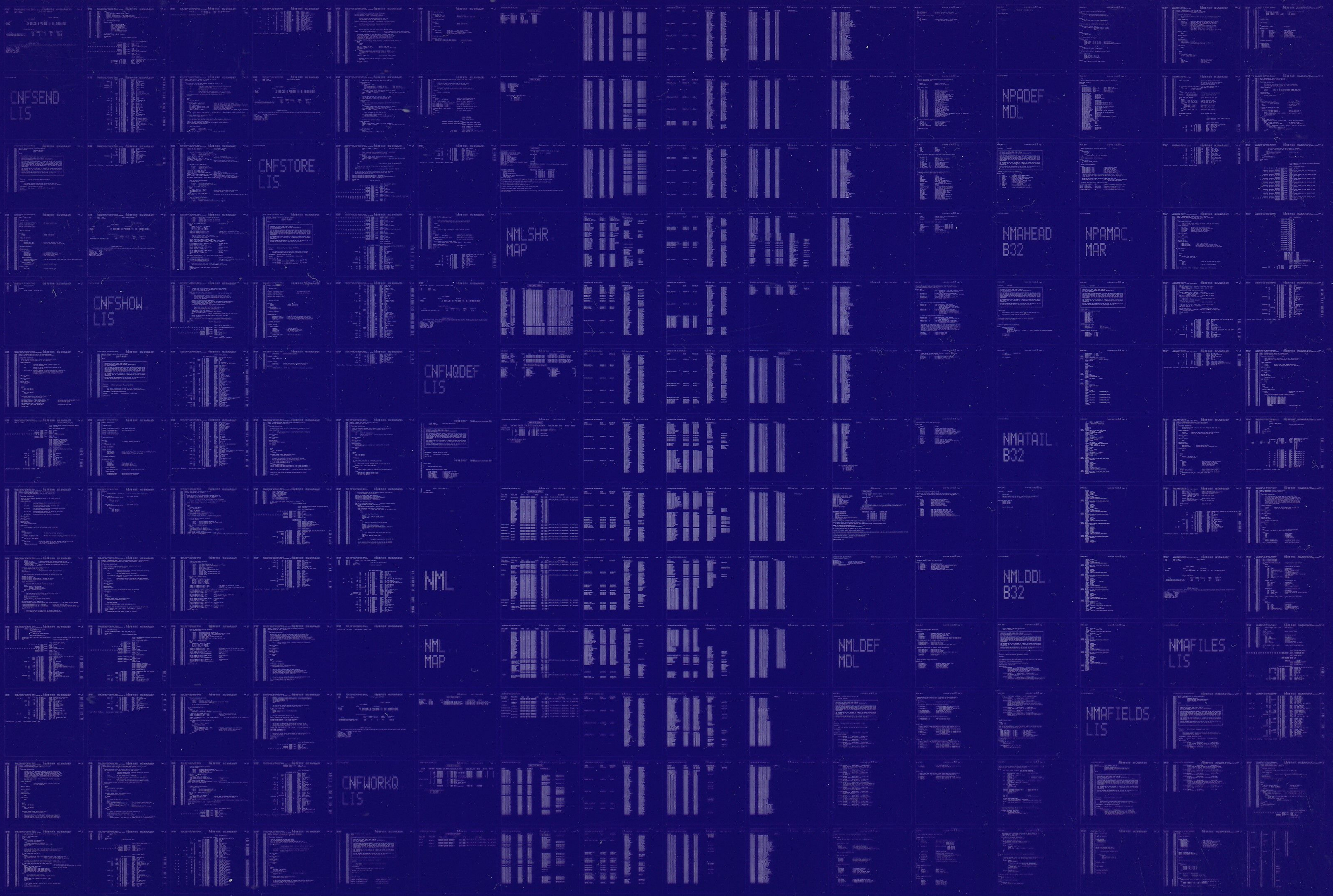
```
                                                00000000'  000B4          .ADDRESS P.AAP                                  ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  000B8 P.AAR:  .ASCII   \TRACE *** ERROR \                     ;
                                                       20  000C7                                                          ;
                                                00000010   000C8 P.AAQ:   .LONG    16                                     ;
                                                00000000'  000CC          .ADDRESS P.AAR                                  ;
    50  4F  4D  20  2D  20  64  69  5F  65  72  6F  74  73  000D0 P.AAT:  .ASCII   \store_id - MOP\                       ;
                                                           000DE          .BLKB    2                                      ;
                                                0000000E   000E0 P.AAS:   .LONG    14                                     ;
                                                00000000'  000E4          .ADDRESS P.AAT                                  ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  000E8 P.AAV:  .ASCII   \TRACE *** ERROR \                     ;
                                                       20  000F7                                                          ;
                                                00000010   000F8 P.AAU:   .LONG    16                                     ;
                                                00000000'  000FC          .ADDRESS P.AAV                                  ;
43  4E  55  46  20  2D  20  64  69  5F  65  72  6F  74  73  00100 P.AAX:  .ASCII   \store_id - FUNCTION LENGTH\           ;
                48  54  47  4E  45  4C  20  4E  4F  49  54  0010F                                                         ;
                                                           0011A          .BLKB    2                                      ;
                                                0000001A   0011C P.AAW:   .LONG    26                                     ;
                                                00000000'  00120          .ADDRESS P.AAX                                  ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  00124 P.AAZ:  .ASCII   \TRACE *** ERROR \                     ;
                                                       20  00133                                                          ;
                                                00000010   00134 P.AAY:   .LONG    16                                     ;
                                                00000000'  00138          .ADDRESS P.AAZ                                  ;
                            64  69  5F  65  72  6F  74  73  0013C P.ABB:  .ASCII   \store_id\                             ;
                                                00000008   00144 P.ABA:   .LONG    8                                      ;
                                                00000000'  00148          .ADDRESS P.ABB                                  ;

                                                                          .EXTRN   SYS$GETTIM

                                                                          .PSECT   $CODE$,NOWRT,2

                                                      0FFC 00000 STORE_ID:
                                                                          .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 0489
                                5B      0000'   CF    9E  00002           MOVAB    P.AAK, R11                             ;
                                5E              08    C2  00007           SUBL2    #8, SP                                 ;
                                                5B    DD  0000A           PUSHL    R11                                    ; 0535
                                        F0      AB    9F  0000C           PUSHAB   P.AAI                                  ; 0534
                                                01    DD  0000F           PUSHL    #1                                     ;
                      0000G     CF              03    FB  00011           CALLS    #3, CNF$TRACE                          ;
                                5A      04      AC    D0  00016           MOVL     CIR, R10                               ; 0542
                                        0A      AA    95  0001A           TSTB     10(R10)                                ;
                                                03    13  0001D           BEQL     1$                                     ;
                                              00E3    31  0001F           BRW      19$                                    ;
                                        04      AE    9F  00022 1$:       PUSHAB   SID                                    ; 0549
                         04     AE              25    D0  00025           MOVL     #37, 4(SP)                             ;
                                        04      AE    9F  00029           PUSHAB   4(SP)                                  ;
                      0000G     CF              02    FB  0002C           CALLS    #2, CNF$GET_ZVM                        ;
                                0E              50    E9  00031           BLBC     STATUS, 2$                             ;
                                56      04      AE    D0  00034           MOVL     SID, R6                                ; 0550
                                16              A6    9F  00038           PUSHAB   22(R6)                                 ;
                  00000000G     00              01    FB  0003B           CALLS    #1, SYS$GETTIM                         ;
                                01              50    E8  00042 2$:       BLBS     STATUS, 3$                             ;
                                                      04  00045           RET                                             ;
                                50      0C      AC    D0  00046 3$:       MOVL     ADRTYPBUF, R0                          ; 0556
            10      A6       06  A0              06    28  0004A           MOVC3    #6, 6(R0), 16(R6)                      ;
                                59      08      AC    D0  00050           MOVL     SYSIDM_DSC, R9                         ; 0567
                    58       04  A9              04    C1  00054           ADDL3    #4, 4(R9), SIMBUF                      ;
                                50              69    3C  00059 4$:       MOVZWL   (R9), R0                               ; 0577
                    50        04   A9  C0 0005C         ADDL2   4(R9), R0                           ;
                    50             58  D1 00060         CMPL    SIMBUF, R0                          ;
                                   16  18 00063         BGEQ    5$                                  ;
                    52             68  3C 00065         MOVZWL  (SIMBUF), TYPE                      ; 0583
                    57        02   A8  9A 00068         MOVZBL  2(SIMBUF), LENGTH                   ; 0584
                                   0F  14 0006C         BGTR    6$                                  ; 0585
                              4C   AB  9F 0006E         PUSHAB  P.AAO                               ; 0589
                              18   AB  9F 00071         PUSHAB  P.AAM                               ; 0588
                                   01  DD 00074         PUSHL   #1                                  ;
             0000G  CF             03  FB 00076         CALLS   #3, CNF$TRACE                       ;
                                   7F  11 0007B 5$:     BRB     18$                                 ; 0587
                    58             03  C0 0007D 6$:     ADDL2   #3, SIMBUF                          ; 0593
                    01             52  D1 00080         CMPL    TYPE, #1                            ; 0605
                                   18  12 00083         BNEQ    8$                                  ;
                    03             57  D1 00085         CMPL    LENGTH, #3                          ; 0607
                                   0B  12 00088         BNEQ    7$                                  ;
                1E  A6             68  B0 0008A         MOVW    (SIMBUF), 30(R6)                    ; 0609
                20  A6        02   A8  90 0008E         MOVB    2(SIMBUF), 32(R6)                   ; 0611
                                   62  11 00093         BRB     17$                                 ; 0607
                              7C   AB  9F 00095 7$:     PUSHAB  P.AAS                               ; 0615
                              64   AB  9F 00098         PUSHAB  P.AAQ                               ; 0614
                                   53  11 0009B         BRB     16$                                 ;
                    02             52  D1 0009D 8$:     CMPL    TYPE, #2                            ; 0624
                                   2B  12 000A0         BNEQ    13$                                 ;
                    02             57  D1 000A2         CMPL    LENGTH, #2                          ; 0629
                                   0F  13 000A5         BEQL    9$                                  ;
                            00B8   CB  9F 000A7         PUSHAB  P.AAW                               ; 0632
                            0094   CB  9F 000AB         PUSHAB  P.AAU                               ; 0631
                                   01  DD 000AF         PUSHL   #1                                  ;
             0000G  CF             03  FB 000B1         CALLS   #3, CNF$TRACE                       ;
                22  A6             68  B0 000B6 9$:     MOVW    (SIMBUF), 34(R6)                    ; 0633
                    58             57  C0 000BA         ADDL2   LENGTH, SIMBUF                      ; 0634
                                   50  D4 000BD         CLRL    INDEX                               ; 0639
      03        22  A6             50  E1 000BF 10$:    BBC     INDEX, 34(R6), 11$                  ; 0640
                              21   A6  96 000C4         INCB    33(R6)                              ; 0641
      F4            50             0F  F3 000C7 11$:    AOBLEQ  #15, INDEX, 10$                     ; 0640
                                   8C  11 000CB 12$:    BRB     4$                                  ; 0599
                    07             52  D1 000CD 13$:    CMPL    TYPE, #7                            ; 0647
                                   07  12 000D0         BNEQ    14$                                 ;
0A    A6            68             06  28 000D2         MOVC3   #6, (SIMBUF), 10(R6)                ; 0649
                                   1E  11 000D7         BRB     17$                                 ; 0650
          00000064  8F             52  D1 000D9 14$:    CMPL    TYPE, #100                          ; 0656
                                   06  12 000E0         BNEQ    15$                                 ;
                24  A6             68  90 000E2         MOVB    (SIMBUF), 36(R6)                    ; 0658
                                   0F  11 000E6         BRB     17$                                 ; 0659
                            00E0   CB  9F 000E8 15$:    PUSHAB  P.ABA                               ; 0669
                            00D0   CB  9F 000EC         PUSHAB  P.AAY                               ; 0668
                                   01  DD 000F0 16$:    PUSHL   #1                                  ;
             0000G  CF             03  FB 000F2         CALLS   #3, CNF$TRACE                       ;
                    58             57  C0 000F7 17$:    ADDL2   LENGTH, SIMBUF                      ; 0670
                                   CF  11 000FA         BRB     12$                                 ; 0577
                            0440   8F  BB 000FC 18$:    PUSHR   #^M<R6,R10>                         ; 0680
             0000V  CF             02  FB 00100         CALLS   #2, INSERT_SID                      ;
                              0C   AC  9F 00105 19$:    PUSHAB  ADRTYPBUF                           ; 0688
                04  AE             0E  D0 00108         MOVL    #14, 4(SP)                          ;
                              04   AE  9F 0010C         PUSHAB  4(SP)                               ;
             0000G  CF             02  FB 0010F         CALLS   #2, CNF$FREE_VM                     ;
```

```
                                        24            50 E9 00114          BLBC    STATUS, 20$                                     ;
                         7E       08    AC            04 C1 00117          ADDL3   #4, SYSIDM_DSC, -(SP)                           ; 0689
                                  04    AE       40   8F 9A 0011C          MOVZBL  #64, 4(SP)                                      ;
                                                 04   AE 9F 00121          PUSHAB  4(SP)                                           ;
                               0000G    CF            02 FB 00124          CALLS   #2, CNF$FREE_VM                                 ;
                                        0F            50 E9 00129          BLBC    STATUS, 20$                                     ;
                                                 08   AC 9F 0012C          PUSHAB  SYSIDM_DSC                                      ; 0690
                                  04    AE            08 D0 0012F          MOVL    #8, 4(SP)                                       ;
                                                 04   AE 9F 00133          PUSHAB  4(SP)                                           ;
                               0000G    CF            02 FB 00136          CALLS   #2, CNF$FREE_VM                                 ;
                                                         04 0013B 20$:     RET                                                     ; 0693

; Routine Size:  316 bytes,     Routine Base:  $CODE$ + 0135
;  513      0694  1 %SBTTL  'insert_sid  Insert SID into CIR block'
;  514      0695  1 ROUTINE INSERT_SID (SID, CIR) =
;  515      0696  1
;  516      0697  1 !++
;  517      0698  1 ! FUNCTIONAL DESCRIPTION:
;  518      0699  1 !
;  519      0700  1 !  Place the system ID message into a doubly linked list of system ID
;  520      0701  1 !  messages in the CIR block, ordered by NI hardware address.
;  521      0702  1 !
;  522      0703  1 ! FORMAL PARAMETERS:
;  523      0704  1 !
;  524      0705  1 !       sid     System ID storage block.
;  525      0706  1 !
;  526      0707  1 !       cir     CIRcuit control block.  Contains buffers and information
;  527      0708  1 !               on the circuit.
;  528      0709  1 !
;  529      0710  1 ! IMPLICIT INPUTS:
;  530      0711  1 !       NONE
;  531      0712  1 !
;  532      0713  1 ! IMPLICIT OUTPUTS:
;  533      0714  1 !       NONE
;  534      0715  1 !
;  535      0716  1 ! ROUTINE VALUE:
;  536      0717  1 ! COMPLETION CODES:
;  537      0718  1 !       Success or a failure returned by a routine which is called
;  538      0719  1 !       from INSERT_SID
;  539      0720  1 !
;  540      0721  1 ! SIDE EFFECTS:
;  541      0722  1 !       NONE
;  542      0723  1 !
;  543      0724  1 !--
;  544      0725  1
;  545      0726  2     BEGIN
;  546      0727  2     MAP
;  547      0728  2         CIR : REF BBLOCK,
;  548      0729  2         SID : REF BBLOCK;
;  549      0730  2
;  550      0731  2     LOCAL
;  551      0732  2         STATUS;
;  552      0733  2
;  553      0734  2
;  554      0735  2     IF .CIR [CIR$L_SIDFLINK] EQL CIR [CIR$L_SIDFLINK]
;  555      0736  2     THEN
;  556      0737  2         !
;  557      0738  2         !   This is the first ID message stored, so just plop it in.
;  558      0739  2         !
;  559      0740  2         INSQUE (.SID, .CIR [CIR$L_SIDBLINK])
;  560      0741  2     ELSE
;  561      0742  3         BEGIN
;  562      0743  3         !
;  563      0744  3         !   Create a doubly linked list ordered by NI Hardware address
;  564      0745  3         !
;  565      0746  3         LOCAL
;  566      0747  3             CMPR_ADR_DSC : BBLOCK [DSC$C_S_BLN],
;  567      0748  3             NSRT_ADR_DSC : BBLOCK [DSC$C_S_BLN],
;  568      0749  3             CMP_SID : REF BBLOCK;
;  569      0750  3
;  3      !
;  3      !     Build a descriptor of the NI hardware address to be used in
;  3      !     a string compare utility routine.
;  3      !
;  3      CH$FILL (0, DSC$C_S_BLN, NSRT_ADR_DSC);
;  3      NSRT_ADR_DSC [DSC$W_LENGTH] = SID$C_ADRLEN;
;  3      NSRT_ADR_DSC [DSC$A_POINTER] = SID [SID$T_HRDWADR];
;  3      CH$FILL (0, DSC$C_S_BLN, CMPR_ADR_DSC);
;  3      CMPR_ADR_DSC [DSC$W_LENGTH] = SID$C_ADRLEN;
;  3
;  3      !
;  3      !     Run thru the whole list
;  3      !
;  3      CMP_SID = .CIR [CIR$L_SIDFLINK];        ! SID being compared against
;  3      WHILE .CMP_SID NEQ CIR [CIR$L_SIDFLINK] DO
;  4          BEGIN
;  4          CMPR_ADR_DSC [DSC$A_POINTER] = CMP_SID [SID$T_HRDWADR];
;  4
;  4          CASE STR$COMPARE (NSRT_ADR_DSC, CMPR_ADR_DSC)
;  4          FROM -1 TO 1 OF     ! Either Less than, Equal to, or Greater than
;  4              SET
;  4
;  4              [-1]:
;  4                  !
;  4                  !   Less than: Insert here
;  4                  !
;  5                  BEGIN
;  5                  INSQUE (.SID, .CMP_SID [SID$L_BLINK]);
;  5                  RETURN TRUE;
;  4                  END;
;  4
;  4              [0] :
;  4                  !
;  4                  !   Equal to: Replace with the new message
;  4                  !
;  5                  BEGIN
;  5                  REMQUE (.CMP_SID, STATUS);
;  5                  INSQUE (.SID, .CMP_SID [SID$L_BLINK]);
;  5                  EXECUTE (CNF$FREE_VM (%REF(SID$C_LENGTH), CMP_SID) );
;  5                  RETURN TRUE;
;  4                  END;
;  4
;  4              [1] :
;  4                  !
;  4                  !   It's greater than: Keep moving down the list
;  4                  !
;  5                  BEGIN
;  5                  CMP_SID = .CMP_SID [SID$L_LINK];
;  4                  END;
;  4
;  4              TES;
;  3          END;                ! End WHILE traversing list of SYSTEM IDs
;  3
;  3      !
;  3      !   It was greater than all the ones in the list so we dropped
;  3      !   down to here where it should be inserted at end of the list
;  3      !
;  627      0808 3          INSQUE (.SID, .CIR [CIR$L_SIDBLINK])
;  628      0809 3
;  629      0810 2          END;                    ! ELSE the list was not empty
;  630      0811 2
;  631      0812 2       RETURN TRUE;
;  632      0813 1       END;                       ! Routine insert_sid



                                             007C 00000 INSERT_SID:
                                                               .WORD    Save R2,R3,R4,R5,R6             ; 0695
                             5E             18 C2 00002        SUBL2    #24, SP                         ;
                             56         08  AC D0 00005        MOVL     CIR, R6                         ; 0735
                             50         40  A6 9E 00009        MOVAB    64(R6), R0                      ;
                             50         40  A6 D1 0000D        CMPL     64(R6), R0                      ;
                                            07 12 00011        BNEQ     1$                              ;
                             44  B6     04  BC 0E 00013        INSQUE   @SID, @68(R6)                   ; 0740
                                            58 11 00018        BRB      5$                              ;
          08          00         6E         00 2C 0001A 1$:    MOVC5    #0, (SP), #0, #8, NSRT_ADR_DSC  ; 0755
                                        08  AE    0001F                                                 ;
                             08  AE         06 B0 00021        MOVW     #6, NSRT_ADR_DSC                ; 0756
                 0C   AE     04  AC         0A C1 00025        ADDL3    #10, SID, NSRT_ADR_DSC+4        ; 0757
          08          00         6E         00 2C 0002B        MOVC5    #0, (SP), #0, #8, CMPR_ADR_DSC  ; 0758
                                        10  AE    00030                                                 ;
                             10  AE         06 B0 00032        MOVW     #6, CMPR_ADR_DSC                ; 0759
                             04  AE     40  A6 D0 00036        MOVL     64(R6), CMP_SID                 ; 0764
                                 52     04  AE D0 0003B 2$:    MOVL     CMP_SID, R2                     ; 0765
                     50     08  AC 00000040 8F C1 0003F        ADDL3    #64, CIR, R0                    ;
                                 50         52 D1 00048        CMPL     R2, R0                          ;
                                            4C 13 0004B        BEQL     8$                              ;
                             14  AE     0A  A2 9E 0004D        MOVAB    10(R2), CMPR_ADR_DSC+4          ; 0767
                                        10  AE 9F 00052        PUSHAB   CMPR_ADR_DSC                    ; 0769
                                        0C  AE 9F 00055        PUSHAB   NSRT_ADR_DSC                    ;
               00000000G         00         02 FB 00058        CALLS    #2, STR$COMPARE                 ;
             02 FFFFFFFF         8F         50 CF 0005F        CASEL    R0, #-1, #2                     ;
               002C            000D      0006     00067 3$:    .WORD    4$-3$,-                         ;
                                                                        6$-3$,-                         ;
                                                                        7$-3$                           ;
                             04  B2     04  BC 0E 0006D 4$:    INSQUE   @SID, @4(R2)                    ; 0778
                                            2E 11 00072 5$:    BRB      9$                              ; 0779
                                 53         62 0F 00074 6$:    REMQUE   (R2), STATUS                    ; 0787
                                 50     04  AE D0 00077        MOVL     CMP_SID, R0                     ; 0788
                             04  B0     04  BC 0E 0007B        INSQUE   @SID, @4(R0)                    ;
                                        04  AE 9F 00080        PUSHAB   CMP_SID                         ; 0789
                             04  AE         25 D0 00083        MOVL     #37, 4(SP)                      ;
                                        04  AE 9F 00087        PUSHAB   4(SP)                           ;
                          0000G  CF         02 FB 0008A        CALLS    #2, CNF$FREE_VM                 ;
                                 10         50 E8 0008F        BLBS     STATUS, 9$                      ;
                                               04 00092        RET                                      ; 0790
                             04  AE         62 D0 00093 7$:    MOVL     (R2), CMP_SID                   ; 0798
                                            A2 11 00097        BRB      2$                              ; 0765
                                 50     08  AC D0 00099 8$:    MOVL     CIR, R0                         ; 0808
                             44  B0     04  BC 0E 0009D        INSQUE   @SID, @68(R0)                   ;
                                 50         01 D0 000A2 9$:    MOVL     #1, R0                          ; 0812
                                               04 000A5        RET                                      ; 0813
```

```
; Routine Size:  166 bytes,    Routine Base:  $CODE$ + 0271
;  634          0814   1 END                                    ! End of module CNFSTORE
;  635          0815   0 ELUDOM




                                                                        .EXTRN  LIB$SIGNAL

;                                      PSECT SUMMARY
;
;       Name                          Bytes                     Attributes
;
;  $PLIT$                                 332  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                                 791  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                                0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)




;                               Library Statistics
;
;                                               -------- Symbols --------       Pages       Processing
;       File                                    Total   Loaded  Percent         Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1            9776       9         0           581         00:01.0
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1            887       2         0            47         00:00.8




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CNFSTORE/OBJ=OBJ$:CNFSTORE MSRC$:CNFSTORE/UPDATE=(ENH$:CNFSTORE)

; Size:          791 code + 332 data bytes
; Run Time:          00:18.2
; Elapsed Time:      00:36.4
; Lines/CPU Min:      2692
; Lexemes/CPU-Min: 19912
; Memory Used:  155 pages
; Compilation Complete
```

0280 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



CNFSEND
LIS

NPADEF
MDL

CNFSTORE
LIS

NMLSHR
MAP

NMAHEAD
B32

NPAMAC
MAR

CNFSHOW
LIS

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS